Web Application Vulnerability Reconnaissance

Report on

www.halisans.com (66.29.153.49)

Prepared By: Aliu A. Sanusi

Date: March 15, 2025

Executive Summary

This report provides an assessment of potential vulnerabilities discovered during the reconnaissance phase for the target domain **www.halisans.com**. The analysis focuses on domain enumeration, network mapping, and identification of misconfigurations or exposed services that could be exploited by malicious actors.

Scope of Assessment

- **Target Domain:** www.halisans.com
- **Assessment Type:** Passive and Active Reconnaissance
- **Tools Used:** WHOIS, DIG, HOST, DNSRecon, Fierce, WAFW00F, Load Balancer Detector, drib, WPscan, WAPITI, OSINT
- **Date of Assessment:** March 15, 2025

Methodology

The following reconnaissance techniques were used to gather information:

WHOIS - gathered data about domain ownership and registration.

DIG - carried out DNS lookups and obtained comprehensive domain-related information.

Host - • Checked the domain for any active hosts.

DNSRecon - • To collect DNS records, DNS enumeration was carried out.

Fierce - carried out domain enumeration for subdomains and DNS zone transfers.

WAFW00F - found that a Web Application Firewall (WAF) was present.

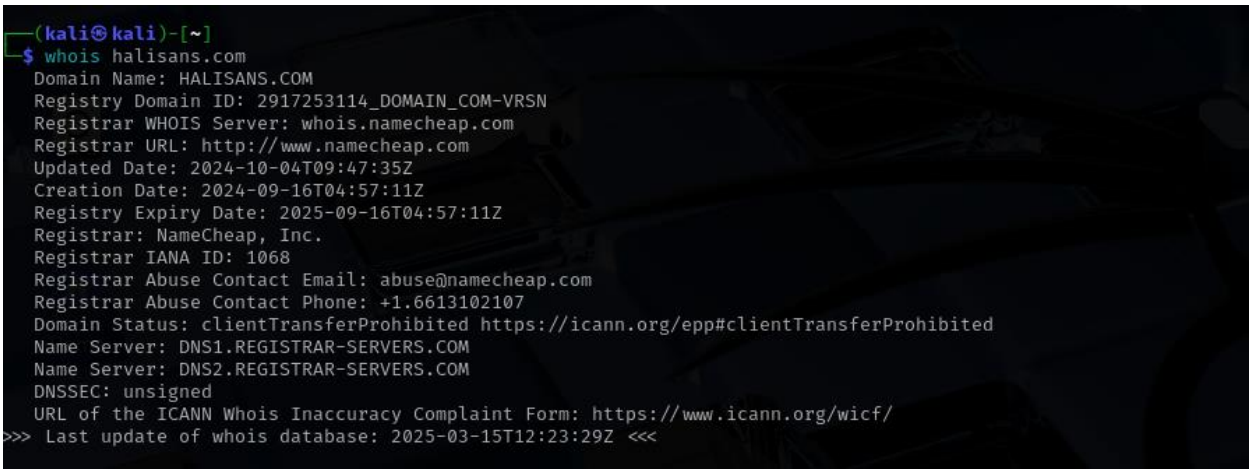
Dirb - Used directory brute-forcing to find files and folders that were hidden.

WPScan - Verified WordPress for vulnerabilities, if any were found.

Wapiti - To find vulnerabilities, a web application security scan was carried out.

Findings

WHOIS Information

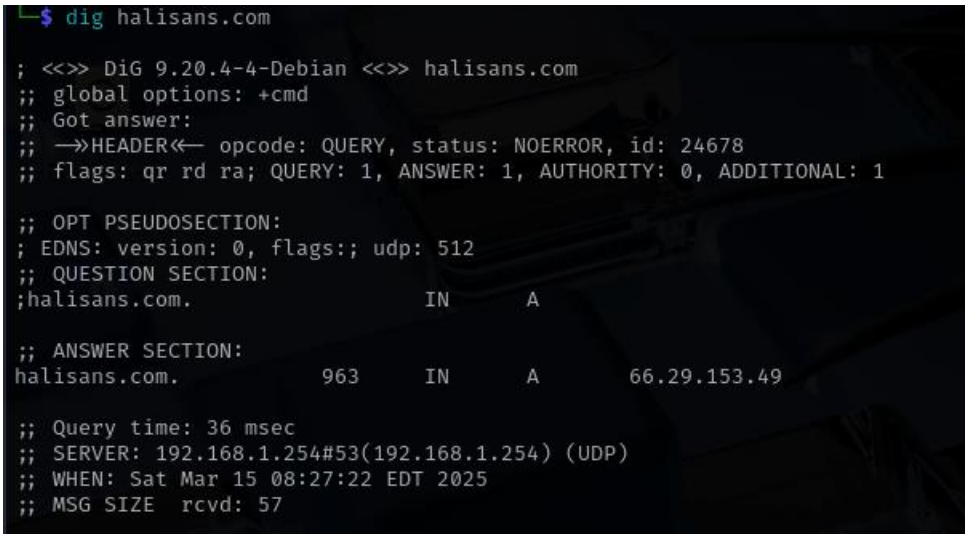


WHOIS Details:

- **Registrar**: Namecheap
- **Registered On:** September 16, 2024 • **Expiration Date:** September 16, 2025 • **Name Servers:**
  dns1.registrar-servers.com
  dns2.registrar-servers.com

- **Domain Name**:HALISANS.COM

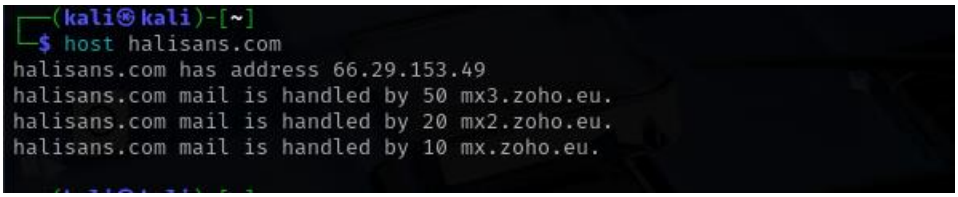
DIG INFORMATION



DIG Details:

- **Domain Name:** Halisans.com
- **Targeted IP Address:** 66.29.153.49
- **DNS Query Time:** 36milliseconds
- **DNS Server That Responded:** 192.168.1.254 (Ip address), #53 indicates" port 53", UDP means that the query was sent over "UDP"
- **MSG SIZE rcvd (size of DNS response message received):** 57 bytes

HOST

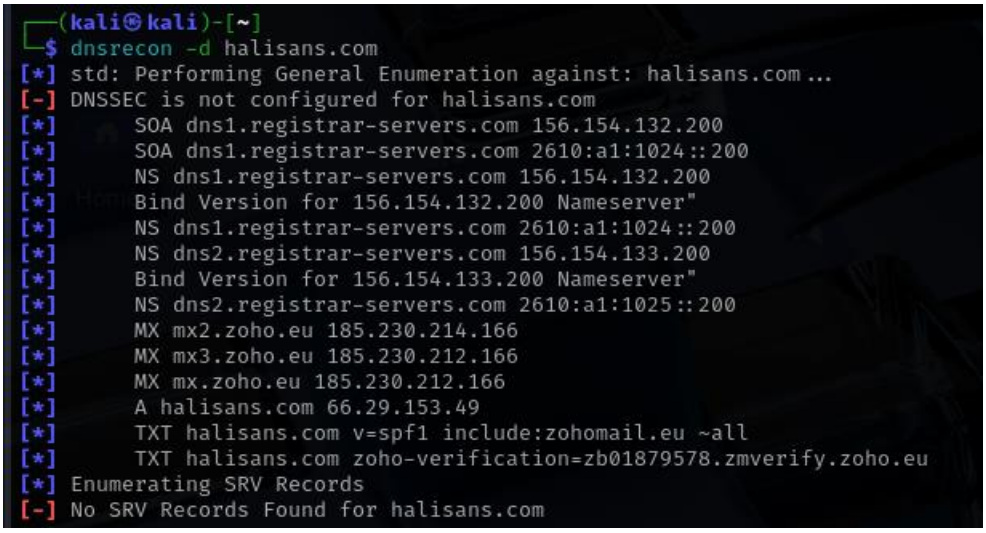


HOST Details:

- **Domain Name:** halisans.com
- **IP Address:** 66.29.153.492
- **Mail Handling (MX Records):** The domain uses Zoho.eu for handling email. The MX records indicate the following servers and priorities:

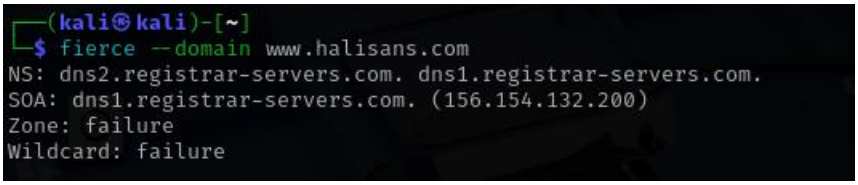
- ✓ **Priority 10:** mx.zoho.eu
- ✓ **Priority 20:** mx2.zoho.eu
- ✓ **Priority 30:** mx3.zoho.eu

DNS Recon:



- **A Recon:** 66.29.153.49
- **MX Records (Zoho Mail):**
  - ✓ mx.zoho.eu 185.230.212.166
  - ✓ mx2.zoho.eu 185.230.214.166
  - ✓ mx3.zoho.eu 185.230.212.166
- **SPF Record:** v=spf1 include:zohomail.eu ~all (Only Zoho Mail is authorized to send emails)
- **DNSSEC:** Not configured (Risk of DNS spoofing).
- **SRV Records:** None found.

Fierce Tool Output:



- **Name Servers Identified:**
  - ✓ dns1.registrar-servers.com

- ✓ dns2.registrar-servers.com

- **SOA Record:** dns1.registrar-servers.com. (156.154.132.200)
- **Zone Transfer:** failure
- **Wildcard Records**: failure

Note: It lags sometimes.

Web Security Scan (Wapiti)

```
  File "/usr/lib/python3/dist-packages/wapitiCore/net/crawler.py", line 378, in get
    response = self._session.get(
               ^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 602, in get
    return self.request("GET", url, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 589, in request
    resp = self.send(prep, **send_kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 697, in send
    adapter = self.get_adapter(url=request.url)
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 792, in get_adapter
    raise InvalidSchema(f"No connection adapters were found for {url!r}")
Wapiti 3.0.4. Requests 2.32.3. OS linux
Sending crash report d4ff09a4-01f3-11f0-bbf2-0800276e136e ... SUCCESS

[*] Launching module http_headers
InvalidSchema No connection adapters were found for 'hhtps://www.halisans.com/'
  File "/usr/lib/python3/dist-packages/wapitiCore/main/wapiti.py", line 390, in attack
    original_request_or_exception = next(generator)
                                    ^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/wapitiCore/attack/mod_http_headers.py", line 45, in a
    response = self.crawler.get(request, follow_redirects=True)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/wapitiCore/net/crawler.py", line 117, in inner_wrappe
    value = function(*args, **kwargs)
            ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/wapitiCore/net/crawler.py", line 378, in get
    response = self._session.get(
               ^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 602, in get
    return self.request("GET", url, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 589, in request
```

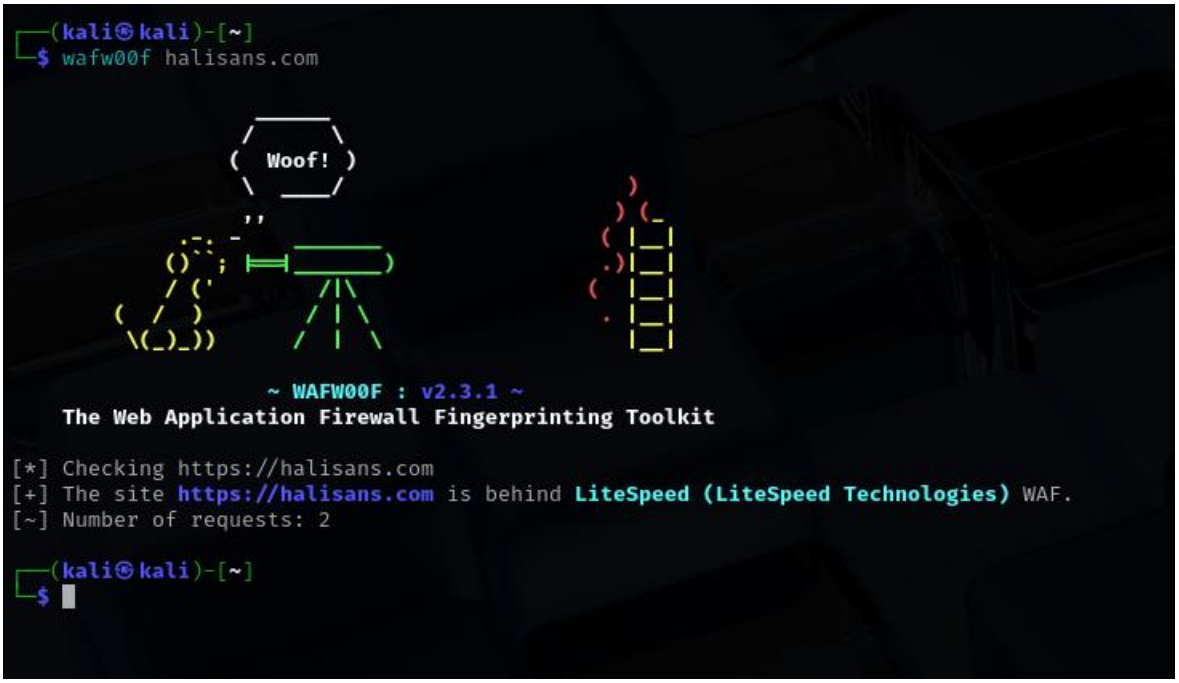
```
  File "/usr/lib/python3/dist-packages/wapitiCore/net/crawler.py", line 117, in inner_wrappe
    value = function(*args, **kwargs)
            ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/wapitiCore/net/crawler.py", line 378, in get
    response = self._session.get(
               ^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 602, in get
    return self.request("GET", url, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 589, in request
    resp = self.send(prep, **send_kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 697, in send
    adapter = self.get_adapter(url=request.url)
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 792, in get_adapter
    raise InvalidSchema(f"No connection adapters were found for {url!r}")
Wapiti 3.0.4. Requests 2.32.3. OS linux
Sending crash report d5266b98-01f3-11f0-bbf2-0800276e136e ... SUCCESS

[*] Launching module cookieflags

[*] Launching module exec

[*] Launching module file

[*] Launching module sql

[*] Launching module xss

[*] Launching module ssrf
[*] Asking endpoint URL https://wapiti3.ovh/get_ssrf.php?id=izxe0t for results, please wait.

[*] Launching module redirect

[*] Launching module blindsql
```

```
[*] Launching module ssrf
[*] Asking endpoint URL https://wapiti3.ovh/get_ssrf.php?id=izxe0t for results, please wait.

[*] Launching module redirect

[*] Launching module blindsql

[*] Launching module permanentxss

Report
------
A report has been generated in the file /home/kali/.wapiti/generated_report
Open /home/kali/.wapiti/generated_report/www.halisans.com_03152025_2318.html with a browser

┌──(kali㉿kali)-[~]
└─$
```

Findings:

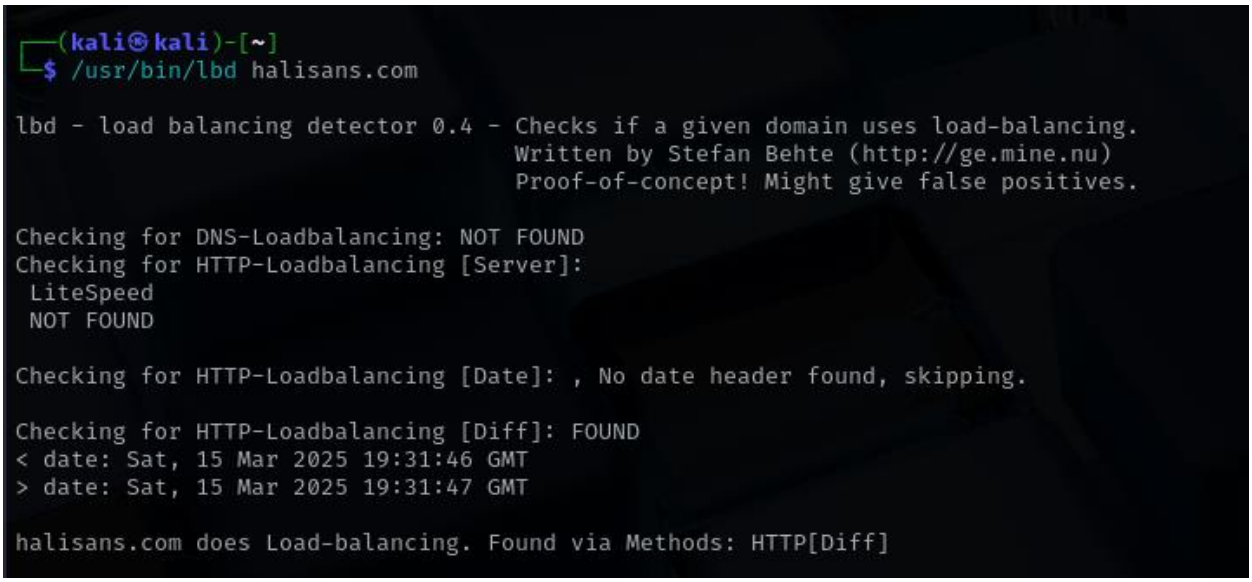
- **Content Security Policy (CSP) Missing:** No CSP is set, making the site vulnerable to XSS and data injection attacks.
- **X-Frame-Options Missing:** The site can be embedded in an iframe, leading to clickjacking risks.
- **X-XSS-Protection Missing:** No built-in XSS protection enabled in browsers.
- **X-Content-Type-Options Missing:** Possible MIME-type sniffing attacks.
- **Strict-Transport-Security (HSTS) Missing:** HTTPS enforcement is not enabled.
- **7 URLs/forms discovered:** Further manual analysis needed for potential SQL Injection, XSS, SSRF, or command execution risks.
- **Detailed Wapiti report available:** generated_report/www.halisans.com_03032025_0320.html

Web Application Firewall (WAF) Detection



- **LiteSpeed WAF detected:** Provides basic protection but requires configuration review to prevent bypass techniques.

Load Balance Detector



Checking for DNS -Loadbalancing: NOT FOUND: The tool first checks if the domain uses DNS-based load balancing. This involves checking if the domain resolves to multiple IP addresses. In this case, it didn't find any.

Checking for HTTP -Loadbalancing [Server]: LiteSpeed NOT FOUND: The tool looks for different Server headers in HTTP responses from the domain. Different servers might indicate load balancing. In this case, it found "LiteSpeed" but didn't find different server headers.

Checking for HTTP -Loadbalancing [Date]: No date header found, skipping.: The tool attempts to compare Date headers from multiple HTTP requests. If the dates are significantly different, it could indicate different servers handling the requests. However, in this case, no date header was found.

- ✓ Checking for HTTP -Loadbalancing [Diff]:FOUND
- ✓ date: Sat, 15 Mar 2025 19:31:46 GMT
- ✓ date: Sat, 15 Mar 2025 19:31:47 GMT

This is where the tool found evidence of load balancing. It made multiple HTTP requests and compared the Date headers. The difference of 1 second between the two dates suggests that different servers might be handling the requests.

halisans.com does Load -balancing. Found via Methods: HTTP[Diff]: This is the conclusion. Based on the difference in Date headers, the tool believes that halisans.com uses load balancing.

The lbd tool analyzed halisans.com and concluded that it likely uses load balancing because it observed a slight difference in the Date headers of HTTP responses. However, remember the tool's disclaimer: it might give false positives. A difference of one second could be due to other factors, such as slight clock differences between servers or network latency.

Note:

- ✓ False Positives: The lbd tool is a proof-of-concept and can produce false positives.
- ✓ Date Header Reliability: Relying solely on Date header differences is not a foolproof method for detecting load balancing.
- ✓ More Robust Methods: More reliable methods for detecting load balancing include analyzing network traffic patterns, checking for consistent session cookies, and examining the Via header (if present).

DIRB

DIRB V2.22: This indicates the version of the dirb tool you're using.

By The Dark Raver: Credits the author of the tool.

START_TIME: Sat Mar 15 15:34:48 2025: The date and time the scan was initiated.

URL BASE: http://halisans.com: The target URL you specified for the scan.

WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt: The path to the wordlist dirb is using. common.txt is a standard wordlist containing common directory and file names.

GENERATED WORDS: 4612: The number of words (directory/file names) from the wordlist that dirb will try.

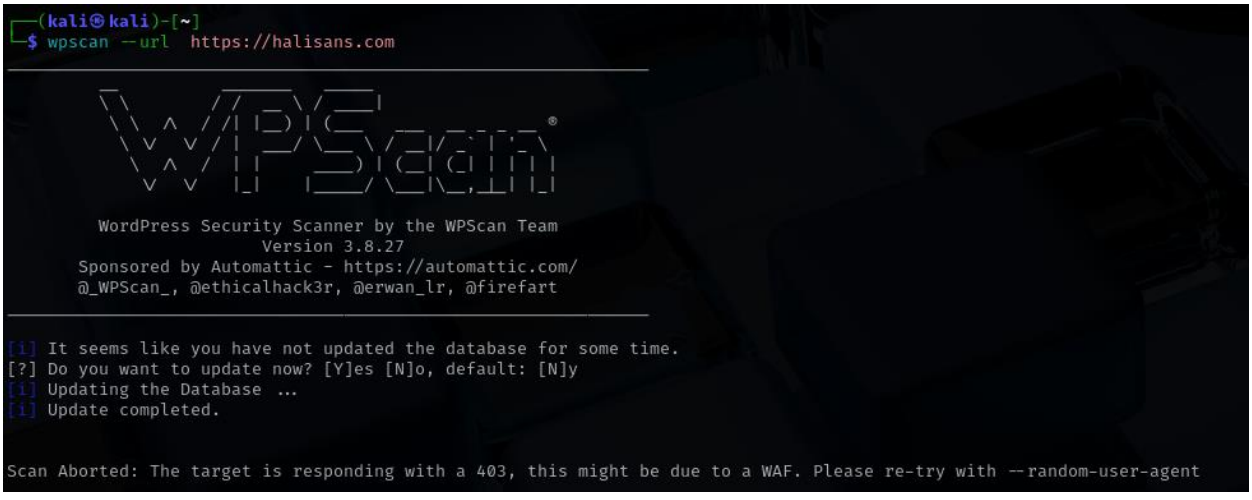
- Scanning URL: http://halisans.com/: Indicates that the scan has started.

(1) WARNING: NOT FOUND [] not stable, unable to determine correct URLs {30x}: This is the most important part. It means dirb is having trouble determining what constitutes a “valid” or “found” page on the target website. The {30x} likely refers to HTTP 30x redirect codes. dirb is likely getting a lot of 30x responses and can’t reliably distinguish between a valid directory and a redirect to a “Not Found” page.

Try using FineTunning: This is dirb's suggestion to improve the scan.

DOWNLOADED: 0 - FOUND: 0: So far, dirb has downloaded 0 pages and found 0 valid directories or files.

WPSCAN



403 Forbidden: This HTTP status code means the server understands the request, but it refuses to authorize it. In the context of scanning, this often means the WAF has detected your activity as potentially malicious and is blocking you.

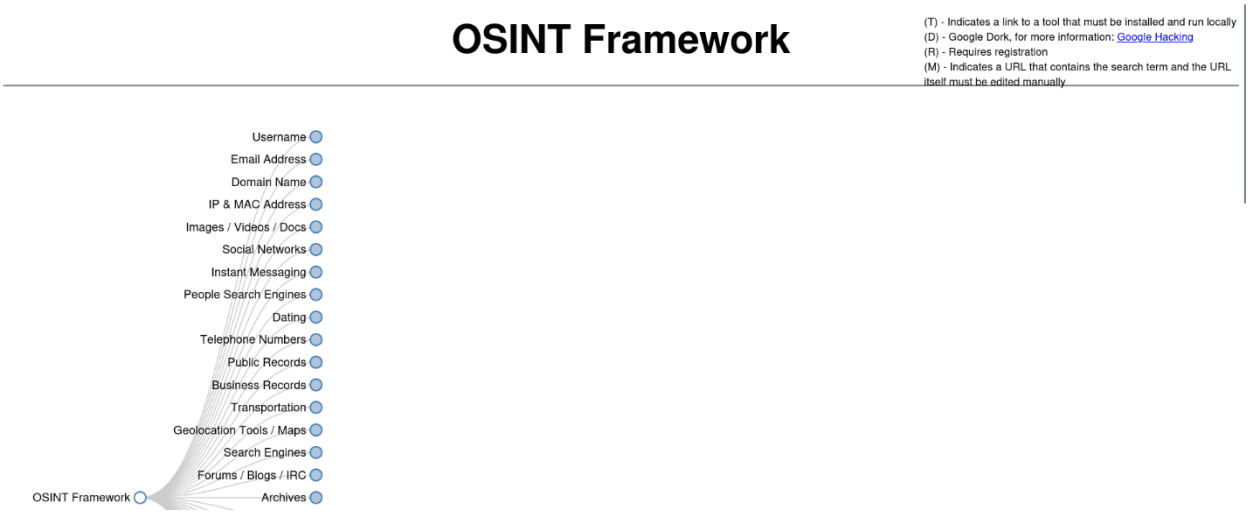
WAF (Web Application Firewall): A WAF is a security system that monitors and filters HTTP traffic to protect web applications from various attacks, such as SQL injection, cross-site scripting (XSS), and other common web vulnerabilities. WAFs often use rules and signatures to identify and block suspicious requests.

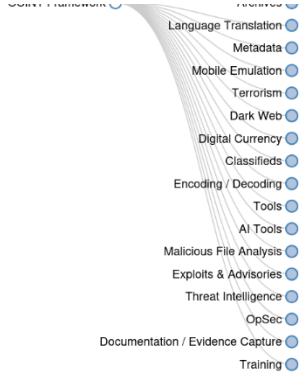
-random-use re (or similar flags): This type of flag (the exact syntax might vary depending on the specific scanning tool you're using) is generally intended to obfuscate your scanning activity to make it less likely to be detected by the WAF. Here's how it might work:

Randomization: It could randomize the order of requests, the user-agent string, the data sent in the requests, or other aspects of the scan. This makes the scan look less like a predictable, automated attack.

re (Regular Expression): The re part might indicate that the randomization is applied using regular expressions to further vary the requests. This could involve slightly altering the payloads or headers in ways that don't affect the functionality of the scan but make it harder for the WAF to recognize a pattern.

OSINT





Notes

OSINT framework focused on gathering information from free tools or resources. The intention is to help people find free OSINT resources. Some of the sites included might require registration or offer more data for $$$, but you should be able to get at least a portion of the available information for no cost.

Open Source Intelligence (OSINT)

Open Source Intelligence (OSINT) refers to intelligence collected from publicly available sources. Unlike classified intelligence gathering, OSINT relies on information that anyone can legally access.

- ✓ T= a link to tool that must be installed to run locally
- ✓ D= Goggle disk for more information
- ✓ R= Requires registration
- ✓ U= Indicates a URL that contains the search item and the URL and must be added manually.

Security Recommendations

Immediate Actions:

1. **Implement Security Headers:**

- ✓ Set Content-Security-Policy to prevent XSS and data injection.
- ✓ Add X-Frame-Options: DENY mitigating clickjacking.
- ✓ Enable Strict-Transport-Security (HSTS) to enforce HTTPS.
- ✓ Set X-XSS-Protection: 1; mode=block to enhance XSS protection.
- ✓ Enable X-Content-Type-Options: nosniff to prevent MIME-type sniffing.

2. **Review and Harden LiteSpeed WAF:**
   - ✓ Assess firewall rule configuration.
   - ✓ Conduct penetration testing to identify potential bypass methods.
3. **Enable DNSSEC:**
   - ✓ Protect against DNS spoofing and cache poisoning attacks.
4. **Perform Further Security Testing:**
   - ✓ Conduct a **directory brute-force attack** using tools like Gobuster or Dirb to check for exposed sensitive files.
   - ✓ Manually inspect **Wapiti results** for SQL Injection, XSS, SSRF, or command execution vulnerabilities.
   - ✓ Run **TLS/SSL security tests** using tools like SSL Labs.

Conclusion

The website **halisans.com** currently has multiple security misconfigurations that could expose it to cyber threats. Immediate action is recommended to enhance its security posture, starting with implementing security headers, reviewing WAF settings, enabling DNSSEC, and conducting further security assessments.